

 **PORTAL**  
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login  
**Search:**  The ACM Digital Library  The Guide  
 +fetch +direction

The ACM Digital Library

 Feedback Report a problem Satisfaction survey

Published before July 2003  
 Terms used fetch direction

Found 5,322 of 139,468

Sort results by relevance  Save results to a Binder  
 Display results expanded form  Search Tips  
 Open results in a new window

Try an Advanced Search  
 Try this search in The ACM Guide

Results 1 - 20 of 200

Result page: 1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

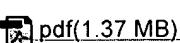
Relevance scale 

### 1 Fetch directed instruction prefetching

Glenn Reinman, Brad Calder, Todd Austin

November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  pdf(1.37 MB)  Additional Information: full citation, abstract, references, citations, index terms  
[Publisher Site](#)

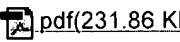
Instruction supply is a crucial component of processor performance. Instruction prefetching has been proposed as a mechanism to help reduce instruction cache misses, which in turn can help increase instruction supply to the processor. In this paper we examine a new instruction prefetch architecture called Fetch Directed Prefetching, and compare it to the performance of next-line prefetching and streaming buffers. This architecture uses a decoupled b ...

### 2 Data streams and time-series: Evaluating continuous nearest neighbor queries for streaming time series via pre-fetching

Like Gao, Zhengrong Yao, X. Sean Wang

November 2002 **Proceedings of the eleventh international conference on Information and knowledge management**

Publisher: ACM Press

Full text available:  pdf(231.86 KB) Additional Information: full citation, abstract, references, citations, index terms

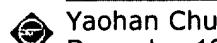
For many applications, it is important to quickly locate the nearest neighbor of a given time series. When the given time series is a streaming one, nearest neighbors may need to be found continuously at all time positions. Such a standing request is called a *continuous nearest neighbor query*. This paper seeks fast evaluation of continuous queries on large databases. The initial strategy is to use the result of one evaluation to restrict the search space for the next. A more fundamental i ...

**Keywords:** continuous query, nearest neighbor, streaming time series

### 3 Predictor-directed stream buffers

Timothy Sherwood, Suleyman Sair, Brad Calder

December 2000 **Proceedings of the 33rd annual ACM/IEEE international symposium**

**on Microarchitecture****Publisher:** ACM PressFull text available:  pdf(187.89 KB) ps(1.12 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)[Publisher Site](#)**4 Direct Execution In A High-Level Computer Architecture** **Yaohan Chu**December 1978 **Proceedings of the 1978 annual conference****Publisher:** ACM PressFull text available:  pdf(870.65 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A high-level computer architecture is one where its structure reflects the constructs of high-level programming languages. This paper describes the structure of a high-level computer architecture, which makes use of the basic concepts of control flow and data flow of programming languages. In this structure, there are the lexical, control and data processors to handle the lexical, control and data elements, respectively. Each processor is associated with an associative memory, and the assoc ...

**Keywords:** Associative memory, Computer architecture, Control processor, Data processor, Direct execution, High-level architecture, Interactive system, Lexical processing

**5 Control Flow Aspects of Semantics-Directed Compiling** **Ravi Sethi**  
October 1983 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 5 Issue 4**Publisher:** ACM PressFull text available:  pdf(1.86 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**6 Design decisions influencing the UltraSPARC's instruction fetch architecture** 

Robert Yung

December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium  
on Microarchitecture****Publisher:** IEEE Computer SocietyFull text available:  pdf(1.35 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Designing a modern microprocessor is a complex task that demands careful balance between cycle time, cycle-per-instruction and area costs. In particular, the instruction fetch unit greatly affects the performance of a multi-issue processor. It must provide adequate bandwidth to sustain peak instruction issue rate and must predict future instruction sequences with high accuracy. In the UltraSPARC prefetch and dispatch unit design, we examined a technique that combined two prediction methods: pred ...

**Keywords:** UltraSPARC, computer architecture, fast cycle time, in-cache prediction, instruction fetch architecture, instruction fetch unit, lower cycle-per-instruction, microprocessor, predictive set-associative cache, prefetch and dispatch unit, trade-off decisions

**7 PL/I Reducer and Direct Processor** 

Masakatsu Sugimoto

 August 1969 **Proceedings of the 1969 24th national conference****Publisher:** ACM PressFull text available:  pdf(828.57 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A computing system is discussed which is aimed at high speed execution of programs written in a problem-oriented language. This computing system has a new machine language. Parallel processing of consecutive instructions and the type of data conversion required for an operation can be easily indicated in the machine language. PL/I is selected for a problem-oriented source language. The programming system in which a source program is translated into the machine language is analyze ...

**8 Stride directed prefetching in scalar processors**  John W. C. Fu, Janak H. Patel, Bob L. Janssens December 1992 **ACM SIGMICRO Newsletter , Proceedings of the 25th annual international symposium on Microarchitecture MICRO 25**, Volume 23 Issue 1-2**Publisher:** IEEE Computer Society Press, ACM PressFull text available:  pdf(1.36 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**9 Next cache line and set prediction**  Brad Calder, Dirk Grunwald May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture ISCA '95**, Volume 23 Issue 2**Publisher:** ACM PressFull text available:  pdf(1.25 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Accurate instruction fetch and branch prediction is increasingly important on today's wide-issue architectures. Fetch prediction is the process of determining the next instruction to request from the memory subsystem. Branch prediction is the process of predicting the likely out-come of branch instructions. Several researchers have proposed very effective fetch and branch prediction mechanisms including branch target buffers (BTB) that store the target addresses of taken branches. An alternative ...

**10 Accelerating shared virtual memory via general-purpose network interface support**  Angelos Bilas, Dongming Jiang, Jaswinder Pal Singh February 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 1**Publisher:** ACM PressFull text available:  pdf(178.88 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

Clusters of symmetric multiprocessors (SMPs) are important platforms for high-performance computing. With the success of hardware cache-coherent distributed shared-memory (DSM), a lot of effort has also been made to support the coherent shared-address-space programming model in software on clusters. Much research has been done in fast communication on clusters and in protocols for supporting software shared memory across them. However, the performance of software virtual memory (SVM) is still ...

**Keywords:** applications, clusters, shared virtual memory, system area networks**11** **Architectural and compiler support for effective instruction prefetching: a cooperative**

 **approach****February 2001 ACM Transactions on Computer Systems (TOCS), Volume 19 Issue 1****Publisher:** ACM PressFull text available:  pdf(432.96 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Instruction cache miss latency is becoming an increasingly important performance bottleneck, especially for commercial applications. Although instruction prefetching is an attractive technique for tolerating this latency, we find that existing prefetching schemes are insufficient for modern superscalar processors, since they fail to issue prefetches early enough (particularly for nonsequential accesses). To overcome these limitations, we propose a new instruction prefetching technique where ...

**Keywords:** compiler optimization, instruction prefetching**12 Data prefetch mechanisms** Steven P. Vanderwiel, David J. Lilja**June 2000 ACM Computing Surveys (CSUR), Volume 32 Issue 2****Publisher:** ACM PressFull text available:  pdf(172.07 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The expanding gap between microprocessor and DRAM performance has necessitated the use of increasingly aggressive techniques designed to reduce or hide the latency of main memory access. Although large cache hierarchies have proven to be effective in reducing this latency for the most frequently used data, it is still not uncommon for many programs to spend more than half their run times stalled on memory requests. Data prefetching has been proposed as a technique for hiding the access lat ...

**Keywords:** memory latency, prefetching**13 Knowledge based approach for the verification of CAD database generated by an automated schematic capture system** J. Y. Tou, W. H. Ki, K. C. Fan, C. L. Huang**October 1987 Proceedings of the 24th ACM/IEEE conference on Design automation****Publisher:** ACM PressFull text available:  pdf(765.41 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

CAD database generated by an automatic schematic capture system needs to be verified before it can be used in design automation. This verification is best performed by a knowledge-based expert system. Presented in this paper is the design of a knowledge-based system for the verification of CAD database generated by AUTORED. Database-driven, pattern-directed inference technique is employed to identify and correct erroneous data records due to misrecognition. This knowledge-based verification ...

**14 A system level perspective on branch architecture performance**

Brad Calder, Dirk Grunwald, Joel Emer

**December 1995 Proceedings of the 28th annual international symposium on Microarchitecture****Publisher:** IEEE Computer Society PressFull text available:  pdf(1.03 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Using network interface support to avoid asynchronous protocol processing in shared virtual memory systems

Angelos Bilas, Cheng Liao, Jaswinder Pal Singh

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture ISCA '99**, Volume 27 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: [pdf\(440.73 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

[Publisher Site](#)

The performance of page-based software shared virtual memory (SVM) is still far from that achieved on hardware-coherent distributed shared memory (DSM) systems. The interrupt cost for asynchronous protocol processing has been found to be a key source of performance loss and complexity. This paper shows that by providing simple and general support for asynchronous message handling in a commodity network interface (NI), and by altering SVM protocols appropriately, protocol activity can be decoupled ...

16 Cache performance of fast-allocating programs

Marcelo J. R. Gonçalves, Andrew W. Appel

October 1995 **Proceedings of the seventh international conference on Functional programming languages and computer architecture**

Publisher: ACM Press

Full text available: [pdf\(1.47 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 The effect of instruction fetch strategies upon the performance of pipelined instruction units

Ramakrishna B. Rau, George E. Rossmann

March 1977 **ACM SIGARCH Computer Architecture News , Proceedings of the 4th annual symposium on Computer architecture ISCA '77**, Volume 5 Issue 7

Publisher: ACM Press

Full text available: [pdf\(562.31 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The interpretation of a machine instruction requires fetching the instruction, decoding the instruction, and then executing it. In addition, if the instruction requires one or more operands, their addresses must be generated and the operands fetched. A large number of processors have been designed to perform some or all of these functions simultaneously on successive instructions. These pipelined processor architectures would appear to permit the decoding of a new instruction each ...

18 Hardware Support: Heads and tails: a variable-length instruction format supporting parallel fetch and decode

Heidi Pan, Krste Asanović

November 2001 **Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems**

Publisher: ACM Press

Full text available: [pdf\(179.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Existing variable-length instruction formats provide higher code densities than fixed-length formats, but are ill-suited to pipelined or parallel instruction fetch and decode. This paper presents a new variable-length instruction format that supports parallel fetch and decode of multiple instructions per cycle, allowing both high code density and rapid execution for high-performance embedded processors. In contrast to earlier schemes that store compressed variable-length instructions in main memory ...

**19 Instruction fetch mechanisms for VLIW architectures with compressed encodings**

Thomas M. Conte, Sanjeev Banerjia, Sergei Y. Larin, Kishore N. Menezes, Sumedh W. Sathaye

December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  pdf(1.34 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

VLIW architectures use very wide instruction words in conjunction with high bandwidth to the instruction cache to achieve multiple instruction issue. This report uses the TINKER experimental testbed to examine instruction fetch and instruction cache mechanisms for VLIWs. A compressed instruction encoding for VLIWs is defined and a classification scheme for i-fetch hardware for such an encoding is introduced. Several interesting cache and i-fetch organizations are described and evaluated through ...

**Keywords:** TINKER experimental testbed, VLIW architectures, compressed encodings, compressed instruction encoding, i-fetch hardware, instruction cache, instruction fetch mechanisms, instruction words, multiple instruction issue, parallel architectures, silo cache, trace-driven simulations

**20 Trace cache: a low latency approach to high bandwidth instruction fetching**

Eric Rotenberg, Steve Bennett, James E. Smith

December 1996 **Proceedings of the 29th annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  pdf(1.38 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As the issue width of superscalar processors is increased, instruction fetch bandwidth requirements will also increase. It will become necessary to fetch multiple basic blocks per cycle. Conventional instruction caches hinder this effort because long instruction sequences are not always in contiguous cache locations. We propose supplementing the conventional instruction cache with a trace cache. This structure caches traces of the dynamic instruction stream, so instructions that are otherwise no ...

**Keywords:** instruction cache, instruction fetching, multiple branch prediction, superscalar processors, trace cache

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)